Lab Project

Milestone III: IR System

Build and evaluate your own IR system using
your topics and relevance assessments.

- ❑ Implement your IR system
  - Training data will be supplied; compute resources available
  - Final system should be deployed to the TIRA platform
- ❑ Evaluate your IR system
  - The previously annotated topics are used for testing
  - Testing is carried out using the TIRA platform
- ❑ Shortly reflect on the assignment in a written report

- ❑ Due Date: 27.11.2023
- ❑ Deliverable: Short reflection (approx. half page), TIRA submission

Task Details

- Input
  - Read the LongEval document corpus from Tira using `ir-datasets` using `tira.third_party_integrations.ir_datasets.load()`
  - Training data: `ir-lab-jena-leipzig-wise-2023/training-20231104-training`
  - Validation data: `ir-lab-jena-leipzig-wise-2023/validation-20231104-training`

- Model ⟹ This is your task!
  - Focus for Milestone III is on *initial retrieval*, i.e., given a corpus produce and initial ranking; fast, reliable, effective scoring based on an index

- Output
  - Write the ranking output to a run file in TREC Run format
  - `https://github.com/joaopalotti/trectools#file-formats`

Tutorials

`https://github.com/webis-de/ir-pad/`

What could you do to improve effectiveness?

- ❑ Retrieval model + its parameters
- ❑ Data cleaning + preprocessing
- ❑ Feature engineering for combined scoring
- ❑ Learning to rank
- ❑ ...

Resources

Example Libraries

- Terrier + pyTerrier (Java + Python Bindings)
  - `https://pyterrier.readthedocs.io/en/latest/terrier-retrieval.html`
  - `http://terrier.org/docs/current/javadoc/org/terrier/matching/models/package-summary.html`
- Anserini + pyserini (Java + Python bindings)
  - `https://github.com/castorini/anserini`
  - `https://github.com/castorini/pyserini`
- Vespa (Dense Indexing)
  - `https://docs.vespa.ai/en/ranking.html`
- Pisa (C++ + Python Bindings)
  - `https://github.com/pisa-engine/pisa`
- Other
  - `https://github.com/textstat/textstat` (Text Features)
  - `https://huggingface.co/models` (Pretrained Models)